Research on Algorithm and VLSI Architecture for HEVC Mode Decision and Reconstruction Loop

Ph.D. student: Heming Sun, Advisor: Shinji Kimura *Waseda University, Japan*

I. Background and Research Purpose

With the development of the information society, multimedia contents are widely used. Video data occupies the majority of multimedia data and it will dramatically grow when high definition (HD) and ultra-HD video applications are popularized in the near future. In order to relieve the burden of video storage and transmission, video compression technique has been widely used. By encoding, large raw video data is compressed to small binary data which is used for storage and transmission. By decoding, the compressed data is decompressed for display. High efficiency video coding (HEVC) is the latest video compression standard which doubles the compression ratio as its predecessor H.264/AVC. However, to reach such high compression ability, many new coding features are adopted and the encoding/decoding complexity becomes 5.2x/2.1x higher than H.264. Therefore, low-complexity algorithms and architectures for HEVC are extremely desired. Mode decision and reconstruction loop are two indispensable components in the video encoding as shown in Fig. 1. Mode decision is used to select the best mode based on the results of the rate-distortion (R-D) cost. The mode with the least R-D cost is selected. Rate represents the number of requiring bits for coding the residual and the best mode information. Distortion stands for the difference between the original picture and the reconstructed picture. After choosing the best mode, the reconstruction loop is carried out to generate the reconstructed pixels for the mode decision afterwards. In HEVC, mode decision and reconstruction loop become much more complex due to two reasons. One reason is the adoption of large transform in HEVC. The largest transform size is 32-point which is 4x larger than H.264. In the state-of-the-art HEVC intra encoder [1], transform consumes about 53% of the overall gate counts. Therefore, the low-cost designs for transform and the system around transform are highly required. The other reason is that there are many more modes in HEVC than H.264. For intra prediction, 5 prediction units (PU) and 35 prediction modes are supported in HEVC. However, there are only 2 PUs and 9 prediction modes in H.264. Therefore, reducing the number of modes requiring the R-D cost is extremely necessary.

In this thesis, we present low-complexity algorithms and architectures for three research topics of the mode decision and reconstruction loop. (1) We give an area-efficient architecture for transform by reusing the calculation results of the butterfly structure and reordering the storing position for the memory organization. (2) We present the system design of the de-quantization and inverse transform. We reuse the four multipliers to save the area consumption and skip the read/write operations of the zero elements for the memories to save the power consumption. (3) We reduce the number of intra modes requiring R-D cost calculation based on the proposed low-complexity cost model. The cost calculation results are reused to save the computation.

II. Architecture for Transform

At first, We give an area-efficient architecture for transform. A complete transform is composed of row transform and column transform which require the logical computational part. In addition, a transpose buffer is required to store the results of row transform. For the logical computational part, We propose a reordered parallel-in serial-out (RPISO) scheme in which the inputs of the butterfly structure could be shared in each clock cycle as shown in Fig. 2. There is an 8-point butterfly structure before the final outputs. In the previous works, $\{X_0, X_1, X_2, X_3\}$ and $\{X_4, X_5, X_6, X_7\}$ are output in two clock cycles, respectively. We reorder the outputs so that $\{X_0, X_3, X_4, X_7\}$ are generated in the first clock cycle. By doing so, X_0 and X_7 can share the inputs of the butterfly structure, so as X_3 and X_4. As a result, the number of calculations could be reduced and 25% normalized gate counts are saved compared with previous works. For the transpose buffer part, static random-access memory (SRAM) instead of register is adopted in order to reduce the area consumption. The proposed data mapping method is shown in Fig. 3 which can achieve 100% I/O utilization of SRAM. By using the proposed mapping method, four results of one column could be fetched in one clock cycle (e.g. [0,0], [1,0], [2,0], [3,0] in Fig. 3). Compared with the existed memories which have the same bit capacity, our memory is much narrower and deeper. As a result, about 62% area consumption can be reduced compared with previous works according to the memory complier results.

III. Architecture for the System of De-quantization and Inverse Transform

Secondly, we present a low-cost system of de-quantization and inverse transform. For the de-quantization, in order to reduce the number of multiplications, we decompose the input coefficients into two parts as shown in Fig. 4. One part is the baseLevel whose value is not greater than 2, thus the multiplication of baseLevel and scaling parameter can be replaced by look-up-tables (LUTs). For the remaining part, the number of positions with non-zero remaining is usually not greater than four in each 4x4 block. Therefore, only four multipliers are provided for processing one 4x4 block in one clock cycle. In the case that there are more than four non-zero remaining, four multipliers are reused in different clock cycles. In order to ensure that the lack of multipliers will not influence the throughput, we analyze the throughput in the worst case with the minimal compression ratio. In addition, in the decoder, since the input coefficients are generated based on the results of syntax elements, we also give a low-delay circuit to produce the inputs of de-quantization. There are three memory operations in the system: read operation of the buffer between de-quantization and inverse transform, write and read operation of the transpose buffer of inverse transform. In order to skip the memory operation in the system, we create a path to detect zero elements as shown in Fig. 5. After the detection, we propose the zero skipping method for the above three operations. As a result, for the de-quantization, 77% normalized area consumption is reduced compared with previous work. For the zero skipping method of the memory part, 29%-86% power consumption can be saved for the memory parts compared with not skipping the memory operations.

IV. Algorithm for Fast Prediction Unit Selection of Intra Prediction

Thirdly, we give a fast prediction unit and prediction mode selection method which is based on a proposed cost model. We first propose a fast preprocessing stage based on a simplified cost model as shown in Fig. 6. After estimating the costs for 8x8, the results are reused to predict the costs for larger PUs. Based on the estimated costs of all the PUs, 2 neighboring PU depths out of 5 are selected to do the R-D cost calculation. To supply PU depth decision with appropriate thresholds, a fast training method is also designed. Still based on the preprocessing results, an efficient mode selection scheme eliminates the necessity to perform fine Hadamard cost calculation in the original HM. We also propose a 32x32 PU compensation scheme to alleviate the mismatch problem of proposed simplified cost and R-D cost due to the lack of large transform size in proposed cost model. The compensation scheme is able to effectively improve coding performance for high-resolution sequences. In comparison with HM, the proposed algorithm achieves about 52% complexity reduction in terms of encoding time, with the corresponding bit rate increment being 1.87%. Moreover, we can achieve stable encoding time reduction for various test sequences as shown in Fig. 7.

V. Conclusion and Future Work

We also implement the R-D cost estimation method in [2]. As a result, for the RDO process and reconstruction loop, overall 1934K gate counts are consumed to meet the throughput of 4K@120fps. In [1], about 807K gate counts are consumed for the RDO process and reconstruction loop, and the throughput is 4K@30fps. About 40% normalized area consumption is reduced compared with [1]. In the future, we will design the system pipeline to improve the hardware efficiency and do the approximate computing design for the large size transform.

[1] G. Pastuszak, and A. Abramowski, "Algorithm and Architecture Design of the H.265/HEVC Intra Encoder," IEEE Transactions on circuits and systems for video technology, vol. 26, no. 1, pp. 210-222, May 2015.

[2] L. Hu, et al., "Hardware-oriented rate-distortion optimization algorithm for HEVC intra-frame encoder," in Proc. of IEEE International Conference in Multimedia & Expo Workshops (ICMEW), pp. 1-6, June 2015.

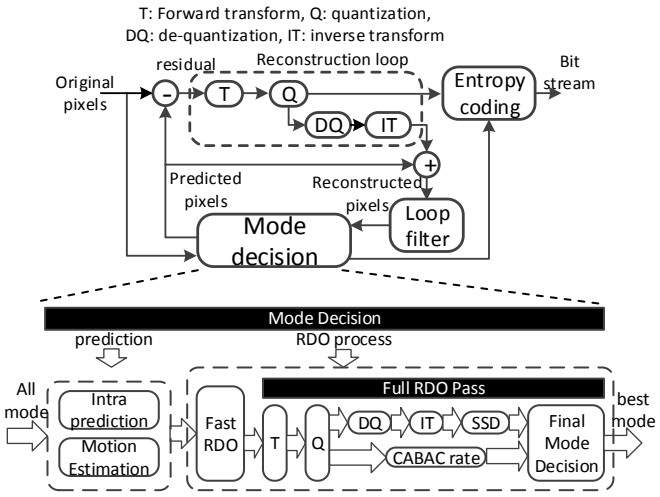

Fig. 1. Encoder diagram.

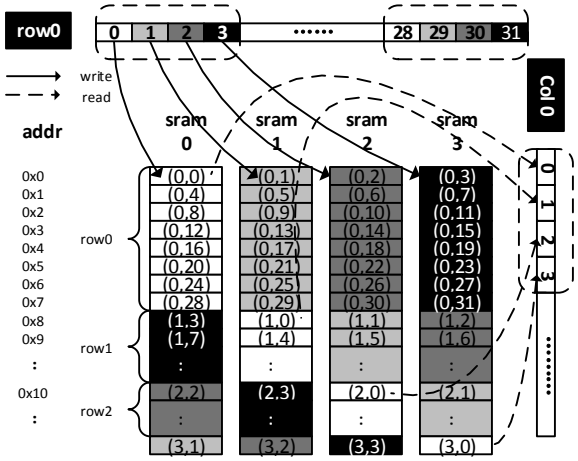

Fig. 3. Proposed data mapping method for transpose buffer part of transform.

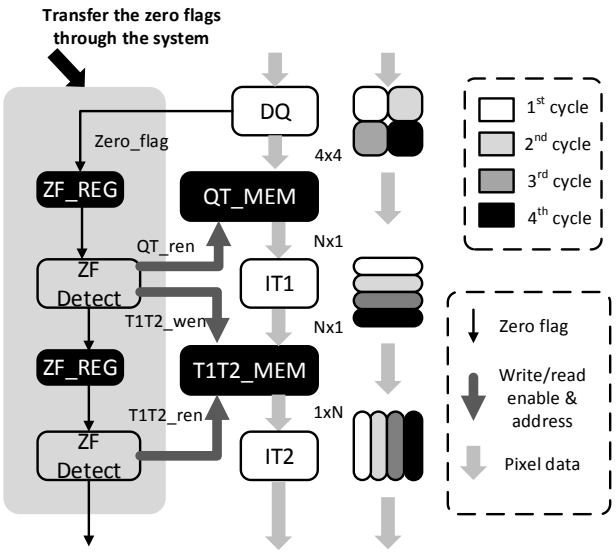

Fig. 5. Proposed system of de-quantization and inverse transform.

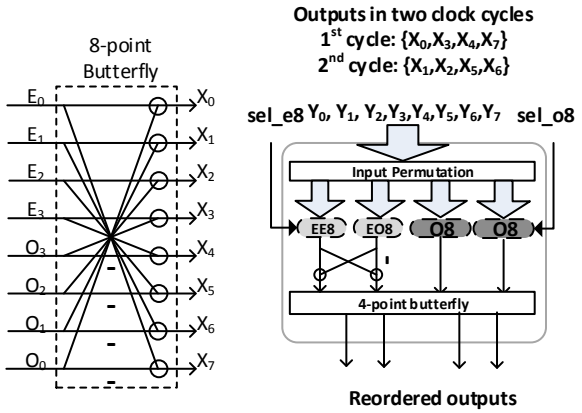

Fig. 2. Reordered parallel-in serial-out (RPISO) scheme for the logical computational part of transform.

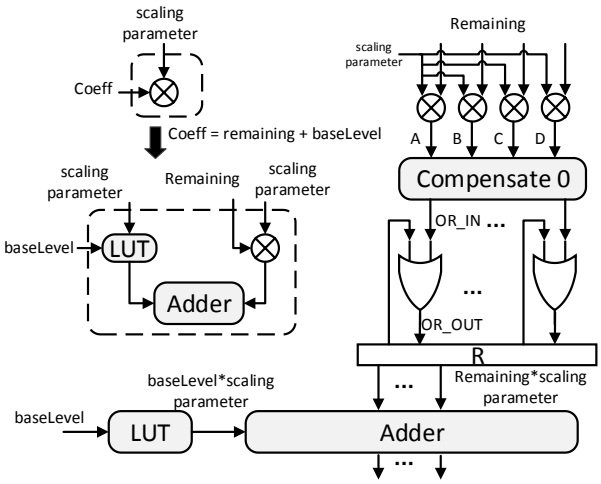

Fig. 4. Proposed architecture for de-quantization.

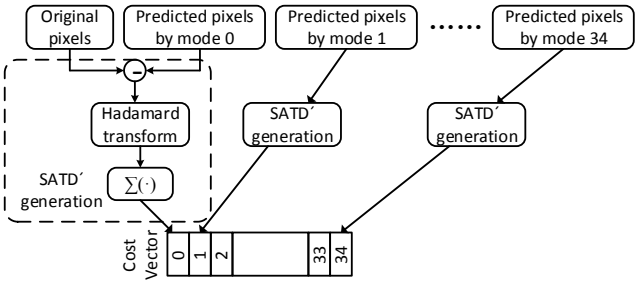

Fig. 6. Proposed simplified cost model for PU 8x8.

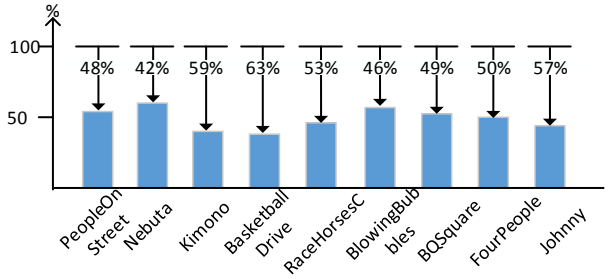

Fig. 7. Encoding time reduction for different test sequences compared with HM.